UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

EX PARTE DEREK B. NOONBURG

APPLICATION NO. 09/901,936

FILING DATE: JULY 9, 2001

BRIEF ON APPEAL

CARR & FERRELL *LLP*
2200 GENG ROAD
PALO ALTO, CA 94303
T: 650.812.3400
F: 650.812.3444

ATTORNEY FOR THE APPELLANT
AND REAL-PARTY-IN-INTEREST

TABLE OF CONTENTS

REAL-PARTY-IN-INTEREST
(37 C.F.R. § 41.37(C)(1)(i))

The real-party-in-interest of U.S. patent application 09/901,936 (the '936 Application) is S3 Graphics Co., Ltd., of the Cayman Islands. An assignment recorded in the US Patent office assigned the inventor's interest to S3 Graphics Co., Ltd. The assignment is recorded at Reel 011990 and Frame 0435 of the U.S. Patent Office's Assignment Division.

RELATED APPEALS AND INTERFERENCES
(37 C.F.R. § 41.37(C)(1)(ii))

The Appellant, the real-party-in-interest, and their undersigned representative are unaware of any related appeals and interferences that are concluded, ongoing, or otherwise prospective as of the date of submission of this BRIEF ON APPEAL.

STATUS OF THE CLAIMS
(37 C.F.R. § 41.37(C)(1)(iii))

Independent claims 1, 19, 21, 24, 25, 26, 27, and 28 are presently pending. Dependent claims 2-18, 20, 22-23, and 29-41 are likewise pending and dependent (either directly or via an intermediate dependent claim) upon one of the aforementioned independent claims. All claims have been (at least) twice rejected. No claims have been allowed or are otherwise objected to by the Examiner.

Claims 1-3, 14, 15, 19 and 24-27 have been rejected under 35 U.S.C. § 102(b) as being anticipated by *Rege*, U.S. Patent No. 5,390,299. Claims 16 and 17 have been rejected under 35 U.S.C. § 103(a) as being obvious over *Rege*. Claims 4, 10, 11, 18 and 20 have been rejected under 35 U.S.C. § 103(a) as being obvious over *Rege* in view of McGuinness, U.S. Patent No. 6,104,416. Claims 5-9 have been rejected under 35 U.S.C. § 103(a) as being obvious over *Rege* and *McGuinness* in view of *Sorin*, U.S. Patent No. 6,631,164. Claim 12 has been rejected under 35 U.S.C. § 103(a) as being obvious over *Rege* in view of Levy, U.S. Patent No. 5,170,251. Claim 13 has been rejected under 35

U.S.C. § 103(a) as being obvious over *Rege* and Levy in view of *McGuinness*. Claims 21-23 and 28-41 have been rejected under 35 U.S.C. § 103(a) as being obvious over *Artieri*, U.S. Patent No. 5,579,052, in view of *Rege*.

The Appellants have elected to appeal only the rejection of independent claims 1, 19, 21, 24, 25, 26, 27, and 28. This election is made for purposes of administrative efficiency of the Board of Patent Appeals and Interferences and to maintain the focus and clarity of argument. This election should not be construed as a concurrence as to the basis for the rejection for any other claim of the '487 Application.

STATUS OF AMENDMENTS (37 C.F.R. § 41.37(C)(1)(iv))

As filed on July 9, 2001, the '936 Application included 41 total claims; claims 1, 19, 21, 24, 25, 26, 27, and 28 were independent. A non-final office action mailed February 2, 2006 indicated the pendency of claims 1-41. In a response dated May 17, 2006, the independent claims 1, 19, 21, 24, 25, 26, 27, and 28 were amended. After another non-final office action of August 10, 2006, containing a new ground of rejection, independent claims 21, 24, 25, 26 and 27 were amended in a response on November 7, 2006.

A final action issued on January 24, 2007. A Request for Continued Examination was filed on April 24, 2007, which amended independent claims 1, 19, 24, 25, 26, 27 and 28, and dependent claim 23. A non-final office action mailed July 27, 2007, indicated the pendency of claims 1-41 and noted entry of the April 24, 2007, response and the pendency of claims 1-41. A response to the July 27 office action was filed October 5, 2007. While this response was labeled an Amendment, no further amendments were made to the claims. Another final action was mailed December 11, 2007, stating that "Applicant's arguments filed October 5, 2007, have been considered but are not persuasive."

Claims 1-41 remain pending. The rejections of independent claims 1, 19, 21, 24, 25, 26, 27, and 28 are appealed herewith.

SUMMARY OF THE CLAIMED SUBJECT MATTER (37 C.F.R. § 41.37(C)(1)(v))[1]

Independent Claim 1

Claim 1 as presented for appeal recites:

A method for generating memory requests to fetch read data from a plurality of locations in a memory, the memory comprising a plurality of memory pages, each of the memory pages having a plurality of words, the method comprising the steps of:

determining the locations of the read data in the memory;

selecting a packetization scheme based on the locations of the read data;

assembling at least one read command for addressing the plurality of locations of the read data; and

fetching the read data from the memory locations and combining it into a plurality of data packets in accordance with the selected packetization scheme, wherein at least one data packet contains data from more than one of the plurality of memory pages.

See infra *CLAIMS APPENDIX*, p. 33.

In the present '936 application, the memory comprises a plurality of memory pages, each page having a plurality of words. Specifically, the memory includes a luminance space and a chrominance space, which each include memory pages having words. "As shown in FIG 5, multiple frame spaces . . . may be allocated in the memory . . . . [e]ach frame space . . . is separated into a luminance (luma) space . . . and a chrominance (chroma) space . . . . The luminance space 410 includes a plurality of 1K memory pages 414 . . . . The chrominance space 412 also includes a plurality of 1K memory pages 416 . . . . The memory 220 is organized in 8-byte words." *SPECIFICATION AS FILED*, p. 14, ¶ 55. "[A] 9x24 byte luminance chunk 800 may fall across 1, 2 or 4 memory pages A-D. The luminance chunk 800 is typically split into one, two or all four of these memory pages in a manner in which page A holds n rows of l data words . . .

[1] All references to the *SPECIFICATION AS FILED* are exemplary and are not intended to be limiting. The present references are made solely to satisfy the requirements of 37 C.F.R. § 41.37(c)(1)(v). No reference is intended—nor should it be construed—as an admission or denial as to any requirement for patentability, including but not limited to those requirements set forth in 35 U.S.C. § 112, ¶ 1 as they pertain to written description and enablement.

page B holds n rows of 3-l data words . . . page C holds 9-n rows of 3-l data words . . . and page D holds 9-n rows of 3-l data words . . . where n = 0, 1, 2 . . . 9 and l = 0, 1, 2 or 3." *SPECIFICATION AS FILED*, p. 18, ¶ 66. "[A] 24x5 byte chrominance chunk 810 may fall across 1, 2 or 4 memory pages E-H. The chrominance chunk 810 is typically split into some or all of these memory pages in a manner in which page E holds m rows of k data words . . . page F holds m rows of 3-k data words . . . page G holds 5-m rows of k data words . . . and page H holds 5-m rows of 3-k data words . . . where m = 0, 1, 2, , . . . , 5 and k = 0, 1, 2 or 3." *SPECIFICATION AS FILED*, p. 18, ¶ 67.

Figure 9 "is a flowchart 900 of a memory request generating process used by the address generator 212." *SPECIFICATION AS FILED*, p. 24, ¶ 77; FIG. 9.

For the method of claim 1, locations of the read data in the memory are first determined. "In step 902, motion vectors pointing to the reference pixel chunks and other relevant parameters such as the picture type, the macroblock type and motion prediction mode . . . are received. *SPECIFICATION AS FILED*, p. 24, ¶ 77. "In step 904, the address generator proceeds with locating the reference pixel chunks." *SPECIFICATION AS FILED*, p. 24, ¶ 78.

Next, a packetization scheme is selected based on the locations of the read data. "[F]or each reference pixel chunk, [the address generator] calculates the values of m, n, l, and k based on the received parameters and the macroblock tiling format used to store reference pictures in memory 220 . . . . [A] packetization scheme is chosen for the specific configuration represented by the values of m, n, l and k." *SPECIFICATION AS FILED*, p. 24, ¶ 78.

At least one read command is then assembled for addressing the plurality of locations of the read data. "The address generator 212 then assembles read commands . . . in step 908 for each pixel chunk based on the chosen packetization scheme." *SPECIFICATION AS FILED*, p. 25, ¶ 79.

Finally, the read data is fetched from the memory locations and combined into a plurality of data packets in accordance with the selected packetization scheme, such that each packet contains data from more than one of the plurality of memory pages. As

indicated above, the luminance and chrominance data are kept on separate pages. *SPECIFICATION AS FILED*, p. 14, ¶ 55. "The read commands . . . are . . . sequenced by the command sequencer 1004. The command sequencer 1004 sends instructions to the memory 220 to request that pixel data corresponding to each reference pixel chunk be sent to the packet assembly unit 1006. . . . The command sequencer 1004 is responsible for generating these sequences of commands for each read, and for retrieving the data as it comes back . . . . The packet assembly unit 1006 assembles read data packets based on the read commands from the command queue." *SPECIFICATION AS FILED*, p. 25, ¶ 80.

Independent Claim 19

Claim 19 as presented for appeal recites:

> A method for packing read data into data packets, the read data being stored in a plurality of locations in a memory, the memory comprising a plurality of memory pages, the method comprising the steps of:
>
> receiving at least one read command requesting the read data, the at least one read command comprising specifications for including in the data packets a plurality of selected portions of the read data from the plurality of memory pages;
>
> sending instructions to the memory according to the at least one read command received, the instructions relating to a manner in which the read data requested is to be obtained from the memory;
>
> receiving the read data from the memory in response to the memory receiving the instructions; and
>
> packing the read data received into the data packets according to the specifications of each of the at least one read commands, wherein at least one data packet contains data from more than one of the plurality of memory pages.

See infra *CLAIMS APPENDIX*, p. 36.

As with claim 1, the memory comprises a plurality of memory pages, each page having a plurality of words. "As shown in FIG 5, multiple frame spaces . . . may be allocated in the memory . . . . Each frame space . . . is separated into a luminance (luma) space . . . and a chrominance (chroma) space . . . . The luminance space 410 includes a plurality of 1K memory pages 414 . . . . The chrominance space 412 also includes a plurality of 1K memory pages 416 . . . . The memory 220 is organized in 8-byte words." *SPECIFICATION AS FILED*, p. 14, ¶ 55. "[A] 9x24 byte luminance chunk 800 may fall across 1, 2 or 4 memory pages A-D. The luminance chunk 800 is typically split into one, two or all four of these memory pages in a manner in which page A holds n rows of l data words . . . page B holds n rows of 3-l data words . . . page C holds 9-n rows of 3-l data words . . . and page D holds 9-n rows of 3-l data words . . . where n = 0, 1, 2 . . . 9 and l = 0, 1, 2 or 3." *SPECIFICATION AS FILED*, p. 18, ¶ 66. "[A] 24x5 byte chrominance chunk 810 may fall across 1, 2 or 4 memory pages E-H. The chrominance chunk 810 is typically split into some or all of these memory pages in a manner in which page E holds m rows

of k data words . . . page F holds m rows of 3-k data words . . . page G holds 5-m rows of k data words . . . and page H holds 5-m rows of 3-k data words . . . where m = 0, 1, 2 . . . 5 and k = 0, 1, 2 or 3." *SPECIFICATION AS FILED*, p. 18, ¶ 67.

First, a read command is received for addressing the plurality of locations of the read data, which includes the specifications for reading the selected data. "The address generator 212 then assembles read commands . . . in step 908 for each pixel chunk based on the chosen packetization scheme. . . . In step 910, the read commands are sent to the interface memory interface unit 224. A read command 232 may include instructions such as putting specific words from specific pages into a data packet." *SPECIFICATION AS FILED*, p. 25, ¶ 79.

Next, instructions are sent to the memory according to the at least one read command received, where the instructions relate to a manner in which the read data requested is to be obtained from memory. "The read commands . . . are . . . sequenced by the command sequencer 1004. The command sequencer 1004 sends instructions to the memory 220 to request that pixel data corresponding to each reference pixel chunk be sent to the packet assembly unit 1006." *SPECIFICATION AS FILED*, p. 25, ¶ 80.

The read data is then received from the memory in response to the memory receiving the instructions. "The command sequencer 1004 is responsible for generating these sequences of commands for each read, and for retrieving the data as it comes back." *SPECIFICATION AS FILED*, p. 25, ¶ 80.

Read data is then packed into data packets according to the specifications of each of the at least one read commands, wherein at least one data packet contains data for more than one of the plurality of memory pages. Each packet contains data from more than one memory page since the luminance and chrominance data are kept on separate pages. *SPECIFICATION AS FILED*, p. 14, ¶ 55. "The command sequencer 1004 is responsible for generating these sequences of commands for each read, and for retrieving the data as it comes back . . . . The packet assembly unit 1006 assembles read data packets based on the read commands from the command queue." *SPECIFICATION AS FILED*, p. 25, ¶ 80.

Independent Claim 21

Claim 21 as presented for appeal recites:

A method for reassembling reference pixel data from a plurality of data packets into a luminance chunk and a chrominance chunk, comprising the steps of:

receiving the plurality of data packets, each data packet comprising a portion of a reference pixel chunk including the luminance chunk and the chrominance chunk;

determining a packetization scheme used to packetize the luminance and chrominance chunks into the plurality of data packets based upon the locations in memory of the data; and

unpacking the plurality of data packets into a reassembled luminance chunk and a reassembled chrominance chunk based on the packetization scheme.

See infra *CLAIMS APPENDIX*, p. 36.

In the method of claim 21, a plurality of data packets are received such that each data packet comprises a portion of a reference pixel chunk which includes a luminance chunk and a chrominance chunk. "The read commands . . . are . . . sequenced by the command sequencer 1004. The command sequencer 1004 sends instructions to the memory 220 to request that pixel data corresponding to each reference pixel chunk be sent to the packet assembly unit 1006. . . . The command sequencer 1004 is responsible for generating these sequences of commands for each read, and for retrieving the data as it comes back . . . The packet assembly unit 1006 assembles read data packets based on the read commands 232 from the command queue, and sends the read data packets to the reference data assembly module 214 (FIG. 2) in the MPEG coprocessor 202 (FIG. 2) through the memory bus 222 (FIG. 2)" *SPECIFICATION AS FILED*, p. 25, ¶ 80. "[A] 9x24 byte luminance chunk 800 may fall across 1, 2 or 4 memory pages A-D. The luminance chunk 800 is typically split into one, two or all four of these memory pages in a manner in which page A holds n rows of l data words." *SPECIFICATION AS FILED*, p. 18, ¶ 66. "[A] 24x5 byte chrominance chunk 810 may fall across 1, 2 or 4 memory pages E-H. The chrominance chunk 810 is typically split into some or all of these memory pages in a

manner in which page E holds m rows of k data words." *SPECIFICATION AS FILED*, p. 18, ¶ 67.

Next, a packetization scheme is determined based on the locations of the read data. "[F]or each reference pixel chunk, [the address generator] calculates the values of m, n, l, and k based on the received parameters and the macroblock tiling format used to store reference pictures in memory 220 . . . . [A] packetization scheme is chosen for the specific configuration represented by the values of m, n, l and k." *SPECIFICATION AS FILED*, p. 24, ¶ 78.

Finally, the plurality of data packets is unpacked into a reassembled luminance chunk and a reassembled chrominance chunk based on the packetization scheme. "The assembly data path module 1204 . . . (3) determines the packetization scheme used to pack the data packets . . . and (4) reassembles the pixel data...into a luminance chunk and a chrominance chunk." *SPECIFICATION AS FILED*, p. 26, ¶ 82.

Independent Claim 24

Claim 24 as presented for appeal recites:

> A computer readable storage medium having embodied thereon a program, the program being executable by a computer processor to perform a method for generating memory requests to fetch read data from a plurality of locations in a memory, the method comprising:
>
> determining the locations of the read data in the memory;
>
> selecting a packetization scheme based on the location of the read data;
>
> assembling at least one read command for addressing the plurality of locations of the read data; and
>
> fetching the read data from the memory locations and combining it into a plurality of data packets in accordance with the selected packetization scheme, wherein at least one data packet contains data from more than one of the plurality of memory pages.

See infra *CLAIMS APPENDIX*, p. 36.

Claim 24 is a computer readable storage medium form of the method of claim 1. The computer readable storage medium has a program which can be executed by a computer processor. "The decoding system 200 includes an MPEG coprocessor 202 for decoding pictures in a compressed video bit stream 204. . . . The MPEG coprocessor . . . make use of a memory 220, and communicates with the memory 220 through a memory bus 222." *SPECIFICATION AS FILED*, p. 9, ¶ 41.

In the method of claim 24, the locations of the read data in the memory are determined. "In step 902, motion vectors pointing to the reference pixel chunks and other relevant parameters such as the picture type, the macroblock type and motion prediction mode . . . are received. *SPECIFICATION AS FILED*, p. 24, ¶ 77. "In step 904, the address generator proceeds with locating the reference pixel chunks." *SPECIFICATION AS FILED*, p. 24, ¶ 78.

Next, a packetization scheme is selected based on the locations of the read data. "[F]or each reference pixel chunk, [the address generator] calculates the values of m, n, l, and k based on the received parameters and the macroblock tiling format used to store reference pictures in memory 220. . . . [A] packetization scheme is chosen for the

specific configuration represented by the values of m, n, l and k." *SPECIFICATION AS FILED*, p. 24, ¶ 78.

At least one read command is assembled for addressing the plurality of locations of the read data. "The address generator 212 then assembles read commands . . . in step 908 for each pixel chunk based on the chosen packetization scheme." *SPECIFICATION AS FILED*, p. 25, ¶ 79.

Finally, the read data is fetched from the memory locations and combined into a plurality of data packets in accordance with the selected packetization scheme, such that each packet contains data from more than one memory page. Each packet contains data from more than one memory page since the luminance and chrominance data are kept on separate pages. *SPECIFICATION AS FILED*, p. 14, ¶ 55. "The read commands . . . are . . . sequenced by the command sequencer 1004. The command sequencer 1004 sends instructions to the memory 220 to request that pixel data corresponding to each reference pixel chunk be sent to the packet assembly unit 1006. . . . The command sequencer 1004 is responsible for generating these sequences of commands for each read, and for retrieving the data as it comes back . . . . The packet assembly unit 1006 assembles read data packets based on the read commands from the command queue." *SPECIFICATION AS FILED*, p. 25, ¶ 80.

Independent Claim 25

Claim 25 as presented for appeal recites:

A computer readable storage medium having embodied thereon a program, the program being executable by a computer processor to perform a method for packing read data into data packets, the method comprising:

receiving at least one read command requesting the read data, the at least one read command comprising specifications for including in the data packets plurality of selected portions of the read data from a plurality of memory pages;

sending instructions to the memory according to the at least one read command received, the instructions relating to a manner in which the read data requested is to be obtained from the memory;

receiving the read data from the memory in response to a memory receiving the instructions; and

packing the read data received into the data packets according to the specifications of each read command, wherein at least one data packet contains data from more than one of the plurality of memory pages.

See infra *CLAIMS APPENDIX*, p. 38.

Claim 25 is relates to a computer readable storage medium. The computer readable storage medium has a program which can be executed by a computer processor. "The decoding system 200 includes an MPEG coprocessor 202 for decoding pictures in a compressed video bit stream 204 . . . . The MPEG coprocessor . . . make use of a memory 220, and communicate with the memory 220 through a memory bus 222." *SPECIFICATION AS FILED*, p. 9, ¶ 41.

The storage medium includes a program that, when executed, performs a method which begins with receiving a read command for addressing the plurality of locations of the read data, which includes the specifications for reading the selected data. "The address generator 212 then assembles read commands . . . in step 908 for each pixel chunk based on the chosen packetization scheme. . . . A read command 232 may include instructions such as putting specific words from specific pages into a data packet." *SPECIFICATION AS FILED*, p. 25, ¶ 79.

Instructions are then sent to the memory according to the at least one read command received, where the instructions relate to a manner in which the read data requested is to be obtained from memory. The read commands are received by the command sequencer, and instructions are sent to the memory indicating how the read data is to be fetched from the memory locations. "The read commands . . . are . . . sequenced by the command sequencer 1004. The command sequencer 1004 sends instructions to the memory 220 to request that pixel data corresponding to each reference pixel chunk be sent to the packet assembly unit 1006." *SPECIFICATION AS FILED*, p. 25, ¶ 80.

The read data is then received from the memory in response to the memory receiving the instructions. "The command sequencer 1004 is responsible for generating these sequences of commands for each read, and for retrieving the data as it comes back." *SPECIFICATION AS FILED*, p. 25, ¶ 80.

Read data is then packed into data packets according to the specifications of each of the at least one read commands, wherein at least one data packet contains data for more than one of the plurality of memory pages. Each packet contains data from more than one memory page since the luminance and chrominance data are kept on separate pages. *SPECIFICATION AS FILED*, p. 14, ¶ 55. "The command sequencer 1004 is responsible for generating these sequences of commands for each read, and for retrieving the data as it comes back . . . . The packet assembly unit 1006 assembles read data packets based on the read commands from the command queue." *SPECIFICATION AS FILED*, p. 25, ¶ 80.

Independent Claim 26

Claim 26 as presented for appeal recites:

> A system for generating memory requests to fetch read data from a plurality of locations in a memory, comprising:
>
> means for determining the locations of the read data in the memory;
>
> means for selecting a packetization scheme based on the locations of the read data and;
>
> means for assembling at least one read command for addressing the plurality of locations of the read data; and
>
> means for fetching the read data from the memory locations and combining it into a plurality of data packets in accordance with the selected packetization scheme, wherein at least one data packet contains data from more than one of the plurality of memory pages.

See infra *CLAIMS APPENDIX*, p. 39.

Claim 26 is a system claim having means plus function elements. The system includes a means for determining locations of the read data in the memory. The system means may be implemented, for example, by address generator 212 (FIG.2). "In step 902, motion vectors pointing to the reference pixel chunks and other relevant parameters such as the picture type, the macroblock type and motion prediction mode . . . are received. *SPECIFICATION AS FILED*, p. 24, ¶ 77. "In step 904, the address generator proceeds with locating the reference pixel chunks." *SPECIFICATION AS FILED*, p. 24, ¶ 78.

The system also includes means that selects a packetization scheme based on the locations of the read data. The selecting system means may also be implemented, for example, by address generator 212 (FIG.2). "[F]or each reference pixel chunk, [the address generator] calculates the values of m, n, l, and k based on the received parameters and the macroblock tiling format used to store reference pictures in memory 220 . . . . [A] packetization scheme is chosen for the specific configuration represented by the values of m, n, l and k." *SPECIFICATION AS FILED*, p. 24, ¶ 78.

Another system means assembles at least one read command for addressing the plurality of locations of the read data. The assembling system means may be implemented by address generator 212 (FIG.2). "The address generator 212 then

assembles read commands . . . in step 908 for each pixel chunk based on the chosen packetization scheme." *SPECIFICATION AS FILED*, p. 25, ¶ 79.

Finally, the system includes a means for fetching read data from the memory locations and combined into a plurality of packets in accordance with the selected packetization scheme, where each packet contains data from more than one memory pages. Each packet contains data from more than one memory page since the luminance and chrominance data are kept on separate pages. *SPECIFICATION AS FILED*, p. 14, ¶ 55. The fetching system means may be implemented by command sequencer 1004. "The read commands . . . are . . . sequenced by the command sequencer 1004. The command sequencer 1004 sends instructions to the memory 220 to request that pixel data corresponding to each reference pixel chunk be sent to the packet assembly unit 1006. . . . The command sequencer 1004 is responsible for generating these sequences of commands for each read, and for retrieving the data as it comes back . . . . The packet assembly unit 1006 assembles read data packets based on the read commands from the command queue." *SPECIFICATION AS FILED*, p. 25, ¶ 80.

Independent Claim 27

Claim 27 as presented for appeal recites:

A system for packing read data into data packets, comprising:

means for receiving at least one read command requesting the read data, the read command comprising specifications for including in the data packets selected portions of the read data from a plurality of memory pages;

means for sending instructions to a memory according to the read command received, the instructions relating to a manner in which the read data requested is to be obtained from the memory;

means for receiving the read data from the memory in response to the memory receiving the instructions; and

means for packing the read data received into the data packets according to the specifications of each read command, wherein at least one data packet contains data from a plurality of memory pages.

See infra *CLAIMS APPENDIX*, p. 40.

Claim 27 is a system having elements recited in means plus function form. The system includes a means for receiving at least one read command for addressing the plurality of locations of the read data, which includes the specifications for reading the selected data. The system means for receiving may be implemented, for example, by interface memory unit 224 (FIG.2). "The address generator 212 then assembles read commands . . . in step 908 for each pixel chunk based on the chosen packetization scheme. . . . In step 910, the read commands are sent to the interface memory unit 224. A read command 232 may include instructions such as putting specific words from specific pages into a data packet." *SPECIFICATION AS FILED*, p. 25, ¶ 79.

The system also includes a means for sending instructions to the memory according to the at least one read command received, where the instructions relate to a manner in which the read data requested is to be obtained from memory. The read commands are received by the command sequencer, and instructions are sent to the memory indicating how the read data is to be fetched from the memory locations. The system means for receiving may be implemented, for example, by command sequencer 1004. "The read commands . . . are . . . sequenced by the command sequencer 1004. The

command sequencer 1004 sends instructions to the memory 220 to request that pixel data corresponding to each reference pixel chunk be sent to the packet assembly unit 1006." *SPECIFICATION AS FILED*, p. 25, ¶ 80.

The system further includes a means for receiving read data from the memory in response to the memory receiving the instructions. The system receiving means may be implemented by command sequencer 1004. "The command sequencer 1004 is responsible for generating these sequences of commands for each read, and for retrieving the data as it comes back." *SPECIFICATION AS FILED*, p. 25, ¶ 80.

Finally, the system includes a means able to pack read data into data packets according to the specifications of each of the at least one read commands, wherein at least one data packet contains data for more than one of the plurality of memory pages. Each packet contains data from more than one memory page since the luminance and chrominance data are kept on separate pages. *SPECIFICATION AS FILED*, p. 14, ¶ 55. The system combining means may be implemented by packet assembly unit 1006. "The command sequencer 1004 is responsible for generating these sequences of commands for each read, and for retrieving the data as it comes back . . . . The packet assembly unit 1006 assembles read data packets based on the read commands from the command queue." *SPECIFICATION AS FILED*, p. 25, ¶ 80.

Independent Claim 28

Claim 28 as presented for appeal recites:

> A system for decoding pictures in a compressed video bit stream, comprising:
>
> a memory having a plurality of memory pages storing reference pixel data;
>
> an address generator coupled to the memory for generating memory commands for fetching the reference pixel data from the memory;
>
> means for packing the fetched reference pixel data into a plurality of data packets according to the specifications of the memory commands;
>
> a reference data assembly module coupled to the address generator for receiving from the memory the plurality of data packets; and
>
> means for unpacking the plurality of data packets and reassembling the fetched reference pixel data into a reassembled video bit stream, wherein at least one data packet contains data from more than one of the plurality of memory pages.

See infra *CLAIMS APPENDIX*, p. 41.

Claim 28 recites a system for decoding pictures in a compressed video bit stream. The system includes a memory having a plurality of memory pages storing reference pixel data. "As shown in FIG 5, multiple frame spaces . . . may be allocated in the memory . . . . Each frame space . . . is separated into a luminance (luma) space . . . and a chrominance (chroma) space . . . . The luminance space 410 includes a plurality of 1K memory pages 414 . . . . The chrominance space 412 also includes a plurality of 1K memory pages 416 . . . . The memory 220 is organized in 8-byte words." *SPECIFICATION AS FILED*, p. 14, ¶ 55. "[A] 9x24 byte luminance chunk 800 may fall across 1, 2 or 4 memory pages A-D. The luminance chunk 800 is typically split into one, two or all four of these memory pages." *SPECIFICATION AS FILED*, p. 18, ¶ 66. "[A] 24x5 byte chrominance chunk 810 may fall across 1, 2 or 4 memory pages E-H. The chrominance chunk 810 is typically split into some or all of these memory pages." *SPECIFICATION AS FILED*, p. 18, ¶ 67.

The system of claim 28 also recites an address generator coupled to the memory generates commands for retrieving the data. "In step 904, the address generator proceeds with locating the reference pixel chunks." *SPECIFICATION AS FILED*, p. 24, ¶ 78.

The system further includes a means for packing the fetched reference pixel data into a plurality of data packets according to the specifications of the memory commands. The specification discloses that the means may be implemented as, for example, a packet assembly unit 1006. "The packet assembly unit 1006 assembles read data packets based on the read commands 232 from the command queue." *SPECIFICATION AS FILED*, p. 25, ¶ 80. "A read command may include instructions such as putting specific words from specific pages into a data packet." Id., p. 25, ¶ 79.

A reference data assembly module is coupled to the address generator and receives data packets from the memory. "FIG. 12 illustrates functional units of the reference data assembly module 214 (FIG. 2). . . . read data packets corresponding to the reference pixel chunk may be received by the reference data assembly module 214." *SPECIFICATION AS FILED*, p. 26, ¶ 82.

Finally, the system includes a means for unpacking the plurality of data packets and reassembling the fetched reference pixel data into a reassembled video bit stream, wherein the at least one data packet contains data from more than one of the plurality of memory pages. The specification discloses that the unpacking means and reassembling means may be implemented, for example, by the assembly data path module 1204. "The assembly data path module 1204 reassembles the pixel data...into a luminance chunk and a chrominance chunk. The pixel data in the reassembled...chunks are buffered." *SPECIFICATION AS FILED*, p. 26, ¶ 82.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL
(37 C.F.R. § 41.37(C)(1)(vi))

(1) Does *Rege* contain all of the elements of independent claims 1, 19, and 24-27 such that it supports a rejection of anticipation under 35 U.S.C. § 102(b)?

(2) Does the combination of *Artieri* and *Rege* render independent claims 21 and 28 obvious under 35 U.S.C. § 103(a)?

ARGUMENT
(37 C.F.R. § 41.37(C)(1)(vii))

I. *THE REGE REFERENCE DOES NOT CONTAIN ALL OF THE ELEMENTS OF CLAIMS 1, 19 AND 24-27*

In the Final Office Action dated December 11, 2007, Examiner rejected independent claims 1, 19 and 24-27 under 35 U.S.C. §102(b) as anticipated by *Rege*, U.S. Patent No. 5,390,299. *Rege* does not support a rejection of claim 1 under 35 U.S.C. §102(b) because the embodiment claimed in claim 1 is not disclosed by *Rege*.

Rege

Rege does not disclose the embodiment of claim 1. *Rege* discloses a system for monitoring the buffer memory occupancy level (how full the buffer is) and providing occupancy level information to a host. In particular, data received from over a network by a host 100 is stored in packet buffer memory 200. A host network adaptor 103 monitors the occupancy of the packet buffer memory and provides this information to a host computer CPU 101. (col. 1, line 63 – col. 2, line 23; Figures 1-2) Congestion avoidance is achieved by monitoring the occupancy of the packet buffer memory.

The pages in pool 300 are used to buffer incoming data packets. Packet buffer memory includes a pages pool 300, a NET receive ring 302, and a HOST transmit ring 303. Each page is 512 bytes in size. Incoming data packets smaller than 512 bytes are contained in one page. Received packets larger than 512 bytes are divided into several pages. (col. 6, lines 13-48)

When a packet is received from the network, the received serial packet data is converted into parallel data words and stored in pages in the packet buffer memory 200. A "start of page" (SOP) pointer and "end of page" (EOP) pointer are created for the stored packets and stored in the NET receive ring 302. (col. 7, lines 1-12) For stored packets to be forwarded to the host, the SOP and EOP pointers are copied from the NET receive ring to the HOST transmit ring 303. (col. 7, lines 15-35) A host bus interface 201 then reads the packet data from the packet buffer memory and writes the data into the

host memory. After all pages in the packet have been delivered to the host, the host bus interface 201 asserts a HOST XMT DONE message and the packet submission to the host is complete. (col. 7, lines 37-48)

Claim 1 is patentably distinguishable from Rege

Claim 1 recites, among other limitations,

> selecting a packetization scheme based on the locations of the read data; and
>
> fetching the read data from the memory locations and combining it into a plurality of data packets in accordance with the selected packetization scheme.

Rege does not disclose "selecting a packetization scheme based on the locations of the read data" as recited in claim 1. *Rege* discloses that incoming packets received from over a network are stored in pages of 512 byes. If an incoming packet is greater than 512 bytes, the packet is stored in multiple pages **as required by the size** of the received packet. As stored packets are sent to a host, a host bus interface retrieves the page(s) of data comprising the packet and sends the complete packet to host memory.

There is no disclosure in *Rege* of "selecting a packetization scheme" as recited in claim 1. Unlike the claimed embodiment, *Rege* discloses one method of grouping pages together that form a packet. There is no disclosure by *Rege* that a "packetization scheme" is affirmatively selected. Rather, *Rege* discloses a single process of building packets from consecutive pages identified by a start page pointer and end page pointer.

Rege further does not disclose selecting a packetization scheme "based on the locations of the read data" as recited in claim 1. *Rege* discloses that a single process is used to form packets based on a start page pointer and an end page pointer. The process involves placing consecutively identified pages in a packet and sending the packet to the host. The process of *Rege* is not selected based on the locations of the pages. Rather, the process is the only process used; though the process accesses data identified in pointers, the data page locations for the data are not used to determine what "scheme" is used to packetize the data.

The portions of *Rege* cited by the Examiner and the Examiner's accompanying arguments do not support the rejection of the "selecting a packatization scheme" element. For example, in support of the rejection to claim 1, Examiner cites to col. 6, lines 12-68, ands states a reasoning as follows:

> Rege teaches generating memory requests to fetch read data from plurality of locations in memory . . determining the locations of read data in the memory (c. 6, ll. 29-35, 57-59). Packets smaller than 512 bytes are stored in one page 305, packets between 512 and 1024 bytes are stored in two pages 305, etc. (c. 6, ll. 18-22.). . . . *[S]o packetization scheme is selected based on how many pages of data need to be read.* Since memory stores plurality of pages (c. 6, ll. 14-18), this means that each page has location in memory. So, *packetization scheme is selected based on locations of read data* (c. 6, ll. 18-22). Rege teaches . . . *combining [the read data] into plurality of data packets in accordance with selected packetization scheme* . . . (c. 6, ll. 12-48, 60-68; c. 7, ll. 1-12, 36-48).

Final Office Action, 7-8 (emphasis added).

Applicant respectfully disagrees with Examiner's interpretation of the scope of *Rege*. Examiner effectively states that because incoming packets may be stored as a single page or several pages based on the size of the packet, the packetizing scheme taught by *Rege* is that packets are stored in a different scheme if they require a different number of pages in buffer memory. Applicant respectfully submits that **the number of pages required to store a packet does not equate to a corresponding packetization scheme**. Rather, the single method used by *Rege* to store incoming data is to divide the incoming packets into sequential 512 byte pages within the buffer memory. Similarly, *Rege* discloses another sole method for rebuilding the data packets from sequential buffer pages and sending them to a host; a data packet rebuilding scheme is not associated with the number pages of data a rebuilt data packet is built from.

To support the rejected element "a packetization scheme is selected based on the locations of the read data," the Examiner cites *Rege* at Col. 6:18-22. The portion cited by the Examiner reads:

> A page 305 is the smallest unit of buffering for packets, so that packets smaller than 512 bytes are stored in one page 305, packets between 512 and 1024 bytes are stored in two pages 305, etc.

Rege at 6:18-22.

A close reading of the cited portion above reveals that *Rege* does not create a packetization scheme by which packets are **created**, but only a way in which packets of a length greater than a memory page length may be divided into page-length portions for storage. Even if dividing a packet into fixed length units could be characterized as a "scheme," *Rege* does not teach selecting such a scheme based upon the locations of the read data, but only based upon the predefined length of pages in the buffer.

In addition to not disclosing the "selecting" element of claim 1, *Rege* also does not disclose "fetching the read data from the memory locations and combining it into a plurality of data packets in accordance with the selected packetization scheme" as recited in claim 1. *Rege* discloses that received packets to be sent from a host are rebuilt from consecutive pages in buffer memory and transmitted. The data packets are not fetched and combined "in accordance with the selected packetization scheme" as claimed. Rather, the data sent to the host is built from consecutive pages marked by pointers. There is no disclosure by *Rege* that the data is fetched or read in accordance with a scheme.

For at least these reasons, *Rege* does not anticipate claim 1.

Claims 19 and 24-27 are patentably distinguishable from Rege

Independent claim 19 similarly recites similar limitations to those of claim 1. For example, claim 19 recites that a read command requesting the read data include specifications for including in the data packets a plurality of *selected portions* of the read data from the plurality of memory pages, and that instructions relating to a manner in which the read data requested is to be obtained from the memory be sent to the memory. As discussed above, *Rege* fails to show packing selected portions of the read data into data packets according to the specifications of a read command or sending instructions about how to retrieve the data, but rather only shows that an *entire* received data packet may be broken into fixed length portions for storage in and retrieval from

the buffer. As such, *Rege* does not disclose selecting a packetization scheme and the read command.

Claims 24-27 also recite one or more patentably distinguishable limitations similar to those contained in claims 1 and 19, and are therefore allowable for the same reasons as discussed above.

II. *THE COMBINATION OF ARTIERI AND REGE DOES NOT RENDER CLAIMS 21 AND 28 OBVIOUS UNDER 35 U.S.C. §103(A)*

The Examiner has rejected independent claims 21 and 28 pursuant to 35 U.S.C. § 103(a) as being unpatentable over *Artieri* in view of *Rege*. *Final Office Action*, 19.

With respect to claim 21, the Examiner admits that:

> Artieri does not teach determining a packetization scheme used to **packetize** the luminance and chrominance chunks into plurality of data packets based upon the locations in memory of the data; unpacking based on packetization scheme.

Final Office Action at 20 (emphasis in original). The Examiner contends that Rege provides such a packetization scheme:

> It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify Artieri to include determining scheme used to packetize luminance and chrominance chunks into data packets based upon locations in memory of data; unpacking based on packetization scheme because Rege teaches if more than 1 page of data needs to be transferred, it can be transferred in 1 packet, so increasing transferring efficiency (c. 6, ll. 18-22).

Final Office Action at 20.

For the reasons set forth above with respect to claim 1, *Rege* does not provide such a packetization scheme, and thus the combination of *Artieri* and *Rege* likewise cannot provide a packetization scheme. Accordingly, claim 21 is allowable over *Artieri* in view of *Rege*.

Similarly to claim 19, claim 28 requires means for packing the fetched reference pixel data into a plurality of data packets according to the specifications of the memory

commands. *Rege* does not teach packing reference pixel data into data packets, but merely receiving data packets and storing them as words in a plurality of memory pages. Further, the "memory commands" of *Rege* are merely commands to retrieve data from specified memory pages and link them together in sequence to recreate the packets, and do not contain any "specifications" for packing the data into packets as that term is used in the present invention. Accordingly, claim 28 is also allowable over *Artieri* in view of *Rege*.

CONCLUSION AND REQUESTED RELIEF

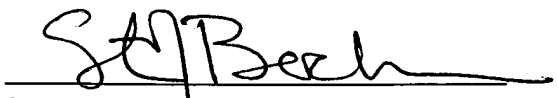
Each of independent claims 1, 19, 21, 24, 25, 26, 27, and 28 of the present application requires that data be combined into a plurality of packets in accordance with a selected packetization scheme. U.S. Patent No. 5,390,299 to *Rege* fails to disclose such a packetization scheme, and the combination of data into packets according to such a scheme. Thus, the Examiner has failed to evidence a prima facie case of anticipation and the 35 U.S.C. § 102(b) rejection is overcome.

In light of the aforementioned, and the fact that the 'other' reference in the Examiner's 35 U.S.C. § 103(a) rejection of independent claims 21 and 28—*Artieri*—also fails to disclose a packetization scheme, much less the combining of data into packets according to such a scheme, the Examiner has failed to evidence a *prima facie* case of obviousness and the 35 U.S.C. § 103(a) rejection is overcome.

In light of the Examiner's failure to disclose each and every element of the presently claimed invention, a *prima facie* case of anticipation or obviousness has not been established. As such, the Examiner's rejection is overcome. The Appellants,

therefore, respectfully request that the final rejection be overturned and the present application remanded with instructions to allow the same.

Respectfully submitted,
Derek B. Noonburg

July 11, 2008 **By:** 

Steve Bachmann, Reg. No. 50,806
CARR & FERRELL *LLP*
2200 GENG ROAD
PALO ALTO, CA 94303
T: 650.812.3400
F: 650.812.3444

ATTORNEY FOR THE APPELLANT
AND REAL-PARTY-IN-INTEREST

CLAIMS APPENDIX
(37 C.F.R. § 41.37(C)(1)(viii))

The claims involved in the present appeal following the April 24, 2007 amendment and October 5, 2007 response submitted in accordance with MPEP § 714(I)(C), 37 C.F.R. § 41.33(a), and 37 C.F.R. § 1.116(b)(2) are as follows:

1. A method for generating memory requests to fetch read data from a plurality of locations in a memory, the memory comprising a plurality of memory pages, each of the memory pages having a plurality of words, the method comprising the steps of:

determining the locations of the read data in the memory;

selecting a packetization scheme based on the locations of the read data;

assembling at least one read command for addressing the plurality of locations of the read data; and

fetching the read data from the memory locations and combining it into a plurality of data packets in accordance with the selected packetization scheme, wherein at least one data packet contains data from more than one of the plurality of memory pages.

2. The method of claim 1 further comprising the step of sending the at least one read command corresponding to the plurality of data packets to the memory.

3. The method of claim 2 further comprising the step of fetching the read data in response to sending the at least one read command.

4. The method of claim 1 wherein the read data comprises a reference pixel chunk having a luminance chunk and a chrominance chunk.

5. The method of claim 4 wherein the step of determining the location of the read data further comprises receiving at least a set of motion vectors pointing to the reference pixel chunk.

6. The method of claim 5 further comprising the step of determining a first set of components associated with the reference pixel chunk based on the at least a set of motion vectors.

7. The method of claim 4 wherein the step of selecting a packetization scheme further comprises combining a part of the luminance chunk and a part of the chrominance chunk into one of the plurality of data packets to be sent from the memory when the luminance chunk overlaps more than one of the plurality of memory pages.

8. The method of claim 4 wherein the step of selecting a packetization scheme further comprises combining a first part of the luminance chunk and a second part of the luminance chunk into one of the plurality of data packets to be sent from the memory when the luminance chunk overlaps more than one of the plurality of memory pages.

9. The method of claim 4 wherein the step of selecting a packetization scheme further comprises combining a first part of the chrominance chunk and a second part of the chrominance chunk into one of the plurality of data packets to be sent from the memory when the chrominance chunk overlaps more than one of the plurality of memory pages.

10. The method of claim 4 further comprising the step of placing a virtual memory page boundary across the luminance chunk, the virtual memory page boundary being associated with the packetization scheme.

11. The method of claim 4 further comprising the step of placing a virtual memory page boundary across the chrominance chunk, the virtual memory page boundary being associated with the packetization scheme.

12. The method of claim 1 wherein the packetization scheme selected maps a first set of components to a second set of components by a table lookup.

13. The method of claim 12 wherein the first set of components comprises the read data corresponding to the luminance chunk and the chrominance chunk, and the second set of components comprises the selected ones of the plurality of words.

14. The method of claim 1 wherein each of the at least one read command includes specifications for combining selected ones of the plurality of words from selected ones of the plurality of memory pages into the plurality of data packets.

15. The method of claim 1 wherein the plurality of data packets is equal to or less than a predetermined number.

16. The method of claim 15 wherein the predetermined number is four and the selected ones of the plurality of memory pages is two.

17. The method of claim 15 wherein the predetermined number is four and the selected ones of the plurality of memory pages is three.

18. The method of claim 1 wherein the plurality of data packets comprise 16 words.

19. A method for packing read data into data packets, the read data being stored in a plurality of locations in a memory, the memory comprising a plurality of memory pages, the method comprising the steps of:

receiving at least one read command requesting the read data, the at least one read command comprising specifications for including in the data packets a plurality of selected portions of the read data from the plurality of memory pages;

sending instructions to the memory according to the at least one read command received, the instructions relating to a manner in which the read data requested is to be obtained from the memory;

receiving the read data from the memory in response to the memory receiving the instructions; and

packing the read data received into the data packets according to the specifications of each of the at least one read commands, wherein at least one data packet contains data from more than one of the plurality of memory pages.

20. The method of claim 19 wherein the read data is a reference pixel chunk comprising a luminance chunk and a chrominance chunk.

21. A method for reassembling reference pixel data from a plurality of data packets into a luminance chunk and a chrominance chunk, comprising the steps of:

receiving the plurality of data packets, each data packet comprising a portion of a reference pixel chunk including the luminance chunk and the chrominance chunk;

determining a packetization scheme used to packetize the luminance and chrominance chunks into the plurality of data packets based upon the locations in memory of the data; and

unpacking the plurality of data packets into a reassembled luminance chunk and a reassembled chrominance chunk based on the packetization scheme.

22. The method of claim 21 further comprising the steps of forming prediction blocks by arranging the plurality of data packets unpacked with any information related to motion vectors, and combining blocks with associated macroblocks to form a reconstructed macroblock.

23. The method of claim 22 further comprising the step of writing the reconstructed macroblock to a memory having a plurality of memory pages;

selecting a packetization scheme based on a location of read data and on fitting the read data into the plurality of data packets; and

assembling at least one read command for fetching the read data from the memory in accordance with the packetization scheme selected, wherein at least one data packet contains data from more than one of the plurality of memory pages.

24. A computer readable storage medium having embodied thereon a program, the program being executable by a computer processor to perform a method for generating memory requests to fetch read data from a plurality of locations in a memory, the method comprising:

determining the locations of the read data in the memory;

selecting a packetization scheme based on the location of the read data;

assembling at least one read command for addressing the plurality of locations of the read data; and

fetching the read data from the memory locations and combining it into a plurality of data packets in accordance with the selected packetization scheme, wherein at least one data packet contains data from more than one of the plurality of memory pages.

25. A computer readable storage medium having embodied thereon a program, the program being executable by a computer processor to perform a method for packing read data into data packets, the method comprising:

receiving at least one read command requesting the read data, the at least one read command comprising specifications for including in the data packets plurality of selected portions of the read data from a plurality of memory pages;

sending instructions to the memory according to the at least one read command received, the instructions relating to a manner in which the read data requested is to be obtained from the memory;

receiving the read data from the memory in response to a memory receiving the instructions; and

packing the read data received into the data packets according to the specifications of each read command, wherein at least one data packet contains data from more than one of the plurality of memory pages.

26. A system for generating memory requests to fetch read data from a plurality of locations in a memory, comprising:

means for determining the locations of the read data in the memory;

means for selecting a packetization scheme based on the locations of the read data and;

means for assembling at least one read command for addressing the plurality of locations of the read data; and

means for fetching the read data from the memory locations and combining it into a plurality of data packets in accordance with the selected packetization scheme, wherein at least one data packet contains data from more than one of the plurality of memory pages.

27. A system for packing read data into data packets, comprising:

means for receiving at least one read command requesting the read data, the read command comprising specifications for including in the data packets selected portions of the read data from a plurality of memory pages;

means for sending instructions to a memory according to the read command received, the instructions relating to a manner in which the read data requested is to be obtained from the memory;

means for receiving the read data from the memory in response to the memory receiving the instructions; and

means for packing the read data received into the data packets according to the specifications of each read command, wherein at least one data packet contains data from a plurality of memory pages.

28. A system for decoding pictures in a compressed video bit stream, comprising:

a memory having a plurality of memory pages storing reference pixel data;

an address generator coupled to the memory for generating memory commands for fetching the reference pixel data from the memory;

means for packing the fetched reference pixel data into a plurality of data packets according to the specifications of the memory commands;

a reference data assembly module coupled to the address generator for receiving from the memory the plurality of data packets; and

means for unpacking the plurality of data packets and reassembling the fetched reference pixel data into a reassembled video bit stream, wherein at least one data packet contains data from more than one of the plurality of memory pages.

29. The system of claim 28 wherein the reference pixel data comprises a luminance chunk and a chrominance chunk.

30. The system of claim 28 wherein the memory commands comprises specifications for combining selected portions of the reference pixel data from a selected one or more of the plurality of memory pages into at least one of the plurality of data packets.

31. The system of claim 28 wherein the reference data assembly module unpacks the plurality of data packets to transform the reference pixel data into a reassembled luminance chunk and a reassembled chrominance chunk.

32. The system of claim 28 wherein the reference data assembly module comprises a plurality of data buffers, each data buffer being configured to receive one of the plurality of data packets.

33. The system of claim 28 wherein the reference data assembly module comprises an additional module for reassembling the reference pixel data based on a set of motion vectors, a table lookup and packetization scheme used to form the plurality of data packets.

34. The system of claim 28 wherein the reference data assembly module comprises a plurality of data buffers for buffering a reassembled luminance chunk and a reassembled chrominance chunk.

35. The system of claim 28 further comprising a variable length decoding module configured to extract a set of motion vectors corresponding to a macroblock in the compressed video bit stream.

36. The system of claim 35 wherein the variable length decoding module sends the extracted set of motion vectors to the address generator.

37. The system of claim 28 further comprising a memory interface unit coupled to the memory.

38. The system of claim 37 wherein the memory interface unit further comprises a memory queue for storing the generated memory commands from the address generator.

39. The system of claim 38 wherein at least one of the plurality of data packets includes the reference pixel data from at least two of the plurality of memory pages based on the generated memory commands in the memory queue.

40. The system of claim 37 wherein the memory interface unit further comprises a sequencer for forwarding the generated memory commands to the memory to obtain the reference pixel data based on specifications.

41. The system of claim 37 wherein the memory interface unit further comprises a packet assembly unit for assembling the plurality of data packets of the reference pixel data obtained from the memory.

EVIDENCE APPENDIX
37 C.F.R. § 41.37(C)(1)(ix)

No evidence submitted pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132 has been presented or entered during prosecution of the present application. As such, no evidence under the aforementioned sections is presented or referenced herewith.

RELATED PROCEEDINGS APPENDIX
37 C.F.R. § 41.37(C)(1)(x)

No related proceedings including appeals or interferences—either concluded, ongoing, or otherwise prospective—are known to the Appellants, real-party-in-interest, nor their agents and representatives. As such, no decisions or documentation related to such a proceedings is presented or referenced herewith.